


```

LL      111111  88888888  FFFFFFFFFF  AAAAAA  000000
LL      111111  88888888  FFFFFFFFFF  AAAAAA  000000
LL      11      88      88  FF      AA      AA  00      00
LL      11      88      88  FF      AA      AA  00      00
LL      11      88      88  FF      AA      AA  00      00
LL      11      88      88  FF      AA      AA  00      00
LL      11      88888888  FFFFFFFF  AA      AA  00      00
LL      11      88888888  FFFFFFFF  AA      AA  00      00
LL      11      88      88  FF      AAAAAAAAAA  00      00
LL      11      88      88  FF      AAAAAAAAAA  00      00
LL      11      88      88  FF      AA      AA  00      00
LL      11      88      88  FF      AA      AA  00      00
LL      11      88888888  FF      AA      AA  000000
LLLLLLLLLLLL  111111  88888888  FF      AA      AA  000000
LLLLLLLLLLLL  111111  88888888  FF      AA      AA  000000

```

```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SSSSSS
LL               II                   SSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```



```
1 0001 0 MODULE LIB$SYS_FAO (
2 0002 0
3 0003 0 IDENT = '1-009' ! File: LIBFAO.B32 EDIT: RKR1009
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY: General Library Support
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine calls $FAOL for the caller, returning the
38 0038 1 resulting string using the semantics of the caller's string.
39 0039 1 The length of the resultant string is limited to 256 bytes max and
40 0040 1 truncation may occur.
41 0041 1
42 0042 1 ENVIRONMENT: VAX-11 User Mode
43 0043 1
44 0044 1 AUTHOR: R. Reichert, CREATION DATE: 20-SEPT-1979
45 0045 1
46 0046 1 MODIFIED BY:
47 0047 1
48 0048 1 1-001 - Original. RKR 20-SEPT-1979
49 0049 1 1-002 - Change name to LIB$SYS_FAO. RKR 15-OCT-79
50 0050 1 1-003 - Change logic for status returned. Improve comments.
51 0051 1 RKR 20-OCT-79
52 0052 1 1-004 - P1 can be a value parameter with value 0, so the
53 0053 1 NULLPARAMETER builtin doesn't work on it. JBS 28-OCT-1979
54 0054 1 1-005 - Improve description of routine. RKR 1-NOV-79
55 0055 1 1-006 - Change STR$ codes to LIB$ codes. RKR 22-JAN-1980
56 0056 1 1-007 - Enhance to recognize additional classes of string descriptors
57 0057 1 on output by always getting data into internal buffer and
```

LIB\$SYS_FAO
1-009

G 9
16-Sep-1984 00:51:03
14-Sep-1984 12:38:47

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBFAO.B32;1

Page 2
(1)

```
: 58      0058 1  | using LIB$SCOPY DXDX6 to deliver to caller's buffer.
: 59      0059 1  | LIB$ANALYZE_SDESC_R3 is used to compute the number of bytes
: 60      0060 1  | actually copied after the copy operation completes.
: 61      0061 1  | Change to use LIB$SCOPY_DXDX6 instead of STR$COPY_DX. This
: 62      0062 1  | avoids having to change STR$ statuses to LIB$ statuses and
: 63      0063 1  | eliminates need for a handler.
: 64      0064 1  | RKR 29-MAY-1981.
: 65      0065 1  | 1-008 - Add special-case code to process string descriptors that
: 66      0066 1  | "read" like fixed string descriptors. RKR 7-OCT-1981.
: 67      0067 1  | 1-009 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to
: 68      0068 1  | LIB$ANALYZE_SDESC_R2. Use LIB$SCOPY_R_DX6 to do copying.
: 69      0069 1  | RKR 18-NOV-1981
: 70      0070 1  | --
: 71      0071 1  |
: 72      0072 1  | !<BLF/PAGE>
```



```

74 0073 1 |
75 0074 1 | SWITCHES:
76 0075 1 |
77 0076 1 |
78 0077 1 | SWITCHES ADDRESSING MODE
79 0078 1 | (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
80 0079 1 |
81 0080 1 |
82 0081 1 | LINKAGES:
83 0082 1 |
84 0083 1 | REQUIRE 'RTLIN:STRLNK'; ! Linkage to LIB$ANALYZE_SDESC_R2
85 0268 1 |
86 0269 1 | TABLE OF CONTENTS:
87 0270 1 |
88 0271 1 |
89 0272 1 | FORWARD ROUTINE
90 0273 1 | LIB$SYS_FAO; ! Formated ASCII output
91 0274 1 |
92 0275 1 |
93 0276 1 | INCLUDE FILES:
94 0277 1 |
95 0278 1 |
96 0279 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Macros for defining psects
97 0374 1 |
98 0375 1 | LIBRARY 'RTLSTARLE'; ! System symbols and macros
99 0376 1 |
100 0377 1 |
101 0378 1 | MACROS:
102 0379 1 |
103 0380 1 | NONE
104 0381 1 |
105 0382 1 | EQUATED SYMBOLS:
106 0383 1 |
107 0384 1 |
108 0385 1 | LITERAL
109 0386 1 | LOCAL_BUF_SIZE = 256;
110 0387 1 |
111 0388 1 |
112 0389 1 | PSECTS:
113 0390 1 |
114 0391 1 | DECLARE_PSECTS (LIB); ! Declare psects for LIB$ facility
115 0392 1 |
116 0393 1 | OWN STORAGE:
117 0394 1 |
118 0395 1 | NONE
119 0396 1 |
120 0397 1 | EXTERNAL REFERENCES:
121 0398 1 |
122 0399 1 |
123 0400 1 | EXTERNAL ROUTINE
124 0401 1 | LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK, ! Extract length
125 0402 1 | ! and address of
126 0403 1 | ! 1st data byte
127 0404 1 | ! from descriptor
128 0405 1 | LIB$COPY_R_DX6 : STRING_JSB ; ! Copy string
```

```
130 0406 1 GLOBAL ROUTINE LIBSSYS_FAO (          ! Formated ASCII output
131 0407 1
132 0408 1     CTRSTR,          ! addr of string desc. for ASCII control string
133 0409 1     OUTLEN,       ! addr of word in which to store output string
134 0410 1                      ! length (Optional)
135 0411 1     OUTBUF,       ! addr of output buffer string descriptor
136 0412 1     P1           ! the first of potentially 17 (max) additional
137 0413 1                      ! parameters.
138 0414 1
139 0415 1     ) =
140 0416 1
141 0417 1 ++
142 0418 1 FUNCTIONAL DESCRIPTION:
143 0419 1
144 0420 1     This routine calls $FAOL for the caller, returning the
145 0421 1     resulting string using the semantics of the caller's string. Since
146 0422 1     output is buffered to an internal buffer which is 256 bytes long,
147 0423 1     the length of the resultant string is limited to 256 bytes max and
148 0424 1     truncation may occur.
149 0425 1
150 0426 1     See $FAO description
151 0427 1
152 0428 1 FORMAL PARAMETERS:
153 0429 1
154 0430 1     CTRSTR.rt.dx      addr of string desc. for ASCII control string
155 0431 1
156 0432 1     OUTLEN.ww.r      addr of word in which to store output string
157 0433 1                      length (Optional parameter)
158 0434 1
159 0435 1     OUTBUF.wt.dx      addr of output buffer string descriptor
160 0436 1
161 0437 1     P1.xx.x          the first of potentially 17 (max)
162 0438 1                      additional parameters. (Optional parameter)
163 0439 1
164 0440 1 For additional description of parameters,
165 0441 1 see $FAO documentation.
166 0442 1
167 0443 1 IMPLICIT INPUTS:
168 0444 1
169 0445 1     NONE
170 0446 1
171 0447 1 IMPLICIT OUTPUTS:
172 0448 1
173 0449 1     NONE
174 0450 1
175 0451 1 COMPLETION CODES:
176 0452 1
177 0453 1     SS$ NORMAL      Procedure successfully completed
178 0454 1 From LIB$COPY R DX6
179 0455 1     LIB$_STRTRU     Success, but source string truncated
180 0456 1     LIB$_INSVIRMEM  Insufficient virtual memory
181 0457 1     LIB$_INVSTRDES  Invalid string descriptor
182 0458 1     LIB$_INVARG     Invalid argument ( usually an unrecognized
183 0459 1                      class of descriptor)
184 0460 1 From $FAOL
185 0461 1     SS$_BUFFEROVF  Successfully completed, but formatted output
186 0462 1                      string overflowed the output buffer and has
```



```
187 0463 1 |
188 0464 1 |      SSS_BADPARAM      been truncated.
189 0465 1 |                      An invalid directive was specified in the FAO
190 0466 1 |                      control string.
191 0467 1 |  SIDE EFFECTS:
192 0468 1 |
193 0469 1 |      NONE
194 0470 1 |
195 0471 1 |  --
196 0472 1 |
197 0473 2 |      BEGIN
198 0474 2 |
199 0475 2 |      BUILTIN
200 0476 2 |          ACTUALCOUNT,
201 0477 2 |          NULLPARAMETER,
202 0478 2 |          ACTUALPARAMETER;
203 0479 2 |
204 0480 2 |      LOCAL
205 0481 2 |          COPY_STATUS,      ! status from buffer copying
206 0482 2 |          STR_STATUS,      ! status from $FAOL call
207 0483 2 |          LOCAL_BUFF : VECTOR [LOCAL_BUF_SIZE, BYTE],      ! local buffer
208 0484 2 |          BUFFER_DESC : BLOCK [8, BYTE];      ! desc for local buffer
209 0485 2 |
210 0486 2 |      MAP
211 0487 2 |          OUTBUF : REF BLOCK [, BYTE],
212 0488 2 |          OUTLEN : REF VECTOR [, WORD];
213 0489 2 |
214 0490 2 |  +
215 0491 2 |  Construct a static descriptor to receive the data from the system
216 0492 2 |  service call
217 0493 2 |  -
218 0494 2 |      BUFFER_DESC [DSC$W_LENGTH] = LOCAL_BUF_SIZE;
219 0495 2 |      BUFFER_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
220 0496 2 |      BUFFER_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
221 0497 2 |      BUFFER_DESC [DSC$A_POINTER] = LOCAL_BUFF;
222 0498 2 |
223 0499 2 |  +
224 0500 2 |  Call $FAOL using the appropriate combination of caller's
225 0501 2 |  parameters, default parameters and local variables
226 0502 2 |  -
227 0503 2 |      STR_STATUS = $FAOL (CTRSTR = ACTUALPARAMETER (1),
228 0504 2 |                          OUTLEN = BUFFER_DESC,
229 0505 2 |                          OUTBUF = BUFFER_DESC,
230 0506 2 |                          PRMLST = (IF (ACTUALCOUNT () LSSU 4)
231 0507 2 |                                  THEN 0 ELSE P1));
232 0508 2 |
233 0509 2 |  +
234 0510 2 |  Copy internal buffer to caller's buffer, letting LIB$SCOPY_R_DX6
235 0511 2 |  worry about its semantics.
236 0512 2 |  -
237 0513 2 |      COPY_STATUS = LIB$SCOPY_R_DX6 (.BUFFER_DESC [DSC$W_LENGTH],
238 0514 2 |                                  .BUFFER_DESC [DSC$A_POINTER],
239 0515 2 |                                  .OUTBUF);
240 0516 2 |
241 0517 2 |  +
242 0518 2 |  If caller supplied optional length parameter, update it.
243 0519 2 |  To do this accurately, we need to inspect the resulting descriptor
244 0519 2 |  as it appears after the copy. Get LIB$ANALYZE_SDESC_R2 to do it.
```

```
244 0520 2 ! No need to check LIB$ANALYZE_SDESC_R2 status. If OUTBUF descr
245 0521 2 was bad, LIB$SCOPY_R_DX6 would have told us about it.
246 0522 2
247 0523 2 IF (NOT NULLPARAMETER(2))
248 0524 2 THEN
249 0525 2 BEGIN
250 0526 2 LOCAL
251 0527 2 OUTBUF_LEN, ! No. of bytes delivered to caller
252 0528 2 OUTBUF_ADDR; ! Address of 1st data byte of callers
253 0529 2 ! buffer
254 0530 2 IF .OUTBUF [DSC$B_CLASS] GTRU DSC$K_CLASS_D
255 0531 2 THEN ! Use generalized length extraction
256 0532 2 BEGIN
257 0533 2 LIB$ANALYZE_SDESC_R2 ( .OUTBUF ; OUTBUF_LEN, OUTBUF_ADDR );
258 0534 2 END
259 0535 2
260 0536 2 ELSE ! Fetch length directly
261 0537 2 OUTBUF_LEN = .OUTBUF [DSC$W_LENGTH] ;
262 0538 2
263 0539 2 OUTLEN [0] = MIN (.BUFFER_DESC [DSC$W_LENGTH], .OUTBUF_LEN) ;
264 0540 2
265 0541 2 END;
266 0542 2
267 0543 2 + At this point we have 2 statuses that we could return to caller.
268 0544 2 Determine which one is the "most meaningful".
269 0545 2
270 0546 2 +
271 0547 2 If original $FAOL call failed, tell him about that
272 0548 2 else give him the copy status.
273 0549 2
274 0550 2 IF NOT .STR_STATUS THEN RETURN (.STR_STATUS)
275 0551 2 ELSE RETURN (.COPY_STATUS) ;
276 0552 2
277 0553 1 END; ! end of LIB$SYS_FAO
```

					.TITLE	LIB\$SYS_FAO	
					.IDENT	\1-009\	
					.EXTRN	LIB\$ANALYZE_SDESC_R2	
					.EXTRN	LIB\$SCOPY_R_DX6	
					.EXTRN	SY\$FAOL	
					.PSECT	_LIB\$CODE,NOWRT, SHR, PIC,2	
					.ENTRY	LIB\$SYS_FAO, Save R2,R3,R4,R5,R6,R7,R8	: 0406
					MOVAB	-260(SP), SP	: 0494
					PUSHL	#17694976	: 0497
					MOVAB	LOCAL_BUFF, BUFFER_DESC+4	: 0507
					CMPB	(AP), #4	:
					BGEQU	1\$:
					CLRL	-(SP)	:
					BRB	2\$:
					MOVAB	P1, R0	:
					PUSHL	R0	:
					PUSHAB	BUFFER_DESC	:
					PUSHAB	BUFFER_DESC	:

						01FC 00000	
					5E	FEFC	CE 9E 00002
					04	010E0100	8F DD 00007
					AE	08	AE 9E 0000D
					04		6C 91 00012
							04 1E 00015
							7E D4 00017
							06 11 00019
					50	10	AC 9E 0001B 1\$:
							50 DD 0001F
					04		AE 9F 00021 2\$:
					08		AE 9F 00024

00000000G	00	04	AC	DD	00027	PUSHL	4(AP)	:
	58		04	FB	0002A	CALLS	#4, SYSSFAOL	:
	57	0C	50	D0	00031	MOVL	R0, STR_STATUS	0514
	52		AC	D0	00034	MOVL	OUTBUF, R7	:
	51		57	D0	00038	MOVL	R7, R2	0512
	50	04	AE	D0	0003B	MOVL	BUFFER_DESC+4, R1	:
			6E	3C	0003F	MOVZWL	BUFFER_DESC, R0	:
	53	00000000G	00	16	00042	JSB	LIB\$COPY_R DX6	:
	02		50	D0	00048	MOVL	R0, COPY_STATUS	:
			6C	91	0004B	CMPB	(AP), #2	0523
			28	1F	0004E	BLSSU	6\$:
		08	AC	D5	00050	TSTL	8(AP)	:
			23	13	00053	BEQL	6\$:
	02	03	A7	91	00055	CMPB	3(R7), #2	0530
			0B	1B	00059	BLEQU	3\$:
	50		57	D0	0005B	MOVL	R7, R0	0533
		00000000G	00	16	0005E	JSB	LIB\$ANALYZE_SDESC_R2	:
			03	11	00064	BRB	4\$	0530
	51		67	3C	00066	MOVZWL	(R7), OUTBUF_LEN	0538
	50		6E	3C	00069	MOVZWL	BUFFER_DESC, R0	0540
	51		50	D1	0006C	CMPL	R0, OUTBUF_LEN	:
			03	15	0006F	BLEQ	5\$:
	50		51	D0	00071	MOVL	OUTBUF_LEN, R0	:
08	BC		50	B0	00074	MOVW	R0, @OUTLEN	:
	04		58	E8	00078	BLBS	STR_STATUS, 7\$	0550
	50		58	D0	0007B	MOVL	STR_STATUS, R0	0551
				04	0007E	RET		:
	50		53	D0	0007F	MOVW	COPY_STATUS, R0	:
			04	00082	RET			0553

; Routine Size: 131 bytes, Routine Base: _LIB\$CODE + 0000

:	278	0554	1	
:	279	0555	1	END
:	280	0556	1	
:	281	0557	0	ELUDOM

! end of module LIBSSYS_FAO

PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	131	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		

LIB\$SYS_FA0
1-009

M 9
16-Sep-1984 00:51:03
14-Sep-1984 12:38:47

VAX-11 Bliss-32 V4.0-742
[LIBRTL.SRC]LIBFA0.B32;1

Page 8
(3)

; _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 10 0 581 00:00.8

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:LIBFA0/OBJ=OBJ\$:LIBFA0 MSRC\$:LIBFA0/UPDATE=(ENH\$:LIBFA0)

; Size: 131 code + 0 data bytes
; Run Time: 00:04.4
; Elapsed Time: 00:17.3
; Lines/CPU Min: 7527
; Lexemes/CPU-Min: 31608
; Memory Used: 72 pages
; Compilation Complete

0206 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY